



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
11.03.1998 Bulletin 1998/11

(51) Int Cl.⁶: **G06F 17/30, G06F 9/44**

(21) Application number: **97306274.8**

(22) Date of filing: **19.08.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC
NL PT SE
 Designated Extension States:
AL LT LV RO SI

- Oketani, Keiichi, c/o Fujitsu Hokuriku Kanazawa-shi, Ishikawa, 920 (JP)
- Matsumoto, Shunji Nakahara-ku, Kawasaki-shi, Kanagawa, 211 (JP)
- Miyahara, Yutaka Nakahara-ku, Kawasaki-shi, Kanagawa, 211 (JP)

(30) Priority: **04.09.1996 JP 234160/96**

(71) Applicant: **FUJITSU LIMITED**
Kawasaki-shi, Kanagawa 211 (JP)

(74) Representative: **Stebbing, Timothy Charles et al**
Haseltine Lake & Co.,
Imperial House,
15-19 Kingsway
London WC2B 6UD (GB)

(72) Inventors:
 • Hayashi, Yoshitaka, c/o Fujitsu Hokuriku Kanazawa-shi, Ishikawa, 920 (JP)

(54) **Intelligent information retrieval program generation system and intelligent information retrieval system**

(57) An intelligent information retrieval program generation system (1) can easily generate an intelligent information retrieval program that is applicable to a wide-area information network such as the Internet. When questions to be used as criteria for determining conclusions, answer alternatives to the questions, and information of conclusions are inputted, a table defining unit (1a) defines a conclusion determining table which comprises answer alternatives to each of the questions and conclusions. A knowledge establishing unit (1b) displays the conclusion determining table on a display screen, receives inputted answer conditions for the con-

clusions to be selected, and establishes the answer conditions in the conclusion determining table. A retrieval program generator (1c) generates an intelligent information retrieval program for decoding the conclusion determining table in which the knowledge has been established, displaying questions, receiving answers to the questions, and determining conclusions depending on the answers. The retrieval program generator (1c) further generates a document containing the intelligent information retrieval program by incorporating the generated intelligent information retrieval program into a document description in the form of hypertext.

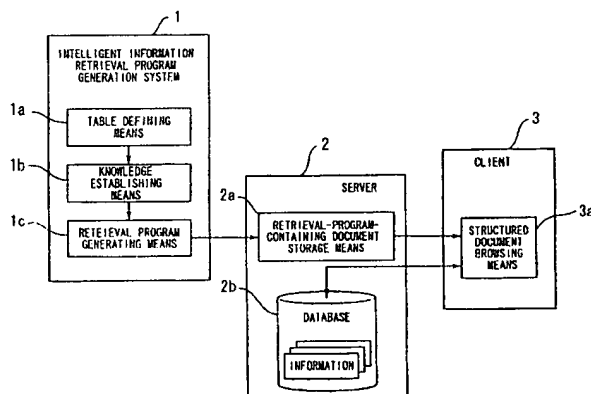


FIG. 1

Description

The present invention relates to an intelligent information retrieval program generation system for generating an intelligent information retrieval program and an intelligent information retrieval system, and more particularly to an intelligent information retrieval program generation system for generating an intelligent information retrieval program incorporated in a document prepared using an information description language that is applicable to a wide-area information network such as the Internet, and an intelligent information retrieval system for retrieving intelligent information on an interactive basis.

As the Internet becomes more and more popular, Internet users can access to various pieces of information from various sources around the world while staying at home or office. On the Internet, an Internet user can browse various documents residing on WWW (World Wide Web) servers in GUI (Graphical User Interface) format using a WWW browser.

The WWW generally operates as follows: When an Internet user inputs an address (URL: Uniform Resource Locator) of information which the user wants to view with a WWW browser, a document having that address is downloaded from a WWW server to the WWW terminal of the user. The WWW browser converts the downloaded document into a predetermined format and then displays it on a display screen. In this manner, the user can view and obtain various pieces of information from WWW servers over the Internet. Most of the information available on the Internet is in the form of documents prepared using an information description language known as HTML (Hyper Text Markup Language).

According to the above WWW process, the Internet user cannot gain access to a desired Web document unless he has an URL of that document. Since the amount of information available on the Internet is huge, the Internet user cannot necessarily reach a desired Web document simply by following links between documents. Therefore, it is necessary for the user to rely on WWW servers which provide document retrieval services. When a retrieval request is received from a WWW terminal, a WWW server starts a retrieval program through a CGI (Common Gateway Interface). The retrieval program makes an inquiry to a database, and converts retrieved data from the database into a document written in the HTML format. The produced HTML document is then transmitted to the WWW terminal.

Heretofore, it has been customary for each WWW server to effect all the processing for retrieving information requested by an Internet user. Therefore, the WWW server has undergone an excessive burden. If WWW terminals can carry out retrieval work for Internet users, then the burden on WWW servers will be reduced. Such a WWW terminal retrieval capability may be incorporated in an intelligent information retrieval system which allows a user to input information relative to a desired

document on an interactive basis and which provides the desired document based on the inputted information. The intelligent information retrieval system with the WWW terminal retrieval capability is not affected by the data transfer rate between a WWW terminal and a WWW server. There has been a demand in the art for a system which can easily generate an information retrieval program that is capable of retrieving information on an interactive basis.

An embodiment of the present invention may provide an intelligent information retrieval program generation system for generating an intelligent information retrieval program, which is applicable to a wide-area information network such as the Internet.

An embodiment of the present invention may also provide an intelligent information retrieval system capable of executing an information retrieval process on an interactive basis at a terminal where desired information is to be received.

There is provided in accordance with the present invention an intelligent information retrieval program generation system for assisting in generating a program for providing an intelligent information retrieval function. The intelligent information retrieval program generation system comprises table defining means for receiving questions to be used as criteria for determining conclusions, answer alternatives to the questions, and information of conclusions, and defining a conclusion determining table which comprises the answer alternatives and the conclusions, knowledge establishing means for displaying the conclusion determining table on a display screen, receiving inputted answer conditions for the conclusions to be selected, and establishing the answer conditions in the conclusion determining table, and retrieval program generating means for generating, based on the conclusion determining table, an intelligent information retrieval program for receiving answers inputted to the questions on an interactive basis and displaying conclusions in response to the answers, and incorporating the intelligent information retrieval program into a document having a logic structure according to a predetermined information description language.

According to the present invention, there is also provided an intelligent information retrieval system for retrieving intelligent information on an interactive basis. The intelligent information retrieval system comprises retrieval-program-containing document storage means for storing a retrieval-program-containing document incorporating an intelligent information retrieval program which has a logic structure according to a predetermined information description language for receiving answers inputted to questions on an interactive basis and displaying conclusions in response to the answers, and structured document browsing means for browsing a document having a logic structure according to the predetermined information description language, and browsing the retrieval-program-containing document and executing the intelligent information retrieval pro-

gram incorporated in the retrieval-program-containing document when the retrieval-program-containing document is acquired from the retrieval-program-containing document storage means.

Reference is made, by way of example, to the accompanying drawings in which:

FIG. 1 is a block diagram showing the principles of the present invention;

FIG. 2 is a block diagram of a network system to which the present invention is applied;

FIG. 3 is a flowchart of a process of generating an information retrieval program;

FIG. 4 is a flowchart of details of a process of defining a table;

FIG. 5 is a flowchart of details of a process of inputting knowledge;

FIG. 6 is a flowchart of details of a process of generating codes;

FIG. 7 is a view of a displayed image of a knowledge editor;

FIG. 8 is a view of a Q editing window;

FIG. 9 is a view of an A editing window;

FIG. 10 is a view of an R editing window;

FIG. 11 is a diagram showing tables generated in an intelligent information retrieval program generation system;

FIG. 12 is a flowchart of a process of generating an intelligent information retrieval program with the knowledge editor;

FIG. 13 is a diagram showing, by way of example, tabular knowledge data that are generated;

FIG. 14 is a diagram showing, by way of example, a document containing an intelligent information retrieval program that has been generated based on tabular knowledge data referred to as "table1";

FIG. 15 is a diagram showing, by way of example, a document containing an intelligent information retrieval program that has been generated based on tabular knowledge data referred to as "table2";

FIG. 16 is a diagram showing, by way of example, an HTML document of a conclusion;

FIG. 17 is a flowchart illustrative of a knowledge structure of the intelligent information retrieval program that has been generated based on the tabular knowledge data referred to as "table1";

FIG. 18 is a flowchart illustrative of a knowledge structure of the intelligent information retrieval program that has been generated based on the tabular knowledge data referred to as "table2";

FIG. 19 is a flowchart illustrative of a knowledge structure of the intelligent information retrieval program that has been generated based on tabular knowledge data referred to as "table3";

FIG. 20 is a view showing, by way of example, a transition from a home page to a document containing the intelligent information retrieval program;

FIG. 21 is a view showing, by way of example, how

an answer is made to "Question 2";

FIG. 22 is a view showing, by way of example, how an answer is made to "Question 3";

FIG. 23 is a view showing, by way of example, a displayed image of a conclusion; and

FIG. 24 is a view showing a displayed image of an announcement which is made when answers are incomplete.

FIG. 1 shows in block form the principles of the present invention. As shown in FIG. 1, an intelligent information retrieval program generation system 1 has a table defining means 1a, a knowledge establishing means 1b, and a retrieval program generating means 1c. When questions to be used as criteria for determining conclusions, answer alternatives to the questions, and information of conclusions are inputted, the table defining means 1a defines a conclusion determining table which comprises answer alternatives to each of the questions and conclusions. The knowledge establishing means 1b displays the conclusion determining table on a display screen, receives inputted answer conditions for the conclusions to be selected, and establishes the answer conditions in the conclusion determining table. The retrieval program generating means 1c generates an intelligent information retrieval program for decoding the conclusion determining table in which the knowledge has been established, displaying questions, receiving answers to the questions, and determining conclusions depending on the answers. The retrieval program generating means 1c further generates a document containing the intelligent information retrieval program by incorporating the generated intelligent information retrieval program into a document description language in the form of hypertext.

The generated document containing the intelligent information retrieval program is stored in a retrieval-program-containing document storage means 2a in a server 2. When the document containing the intelligent information retrieval program is accessed by a structured document browsing means 3a in a client 3 which is connected to the server 2, the document containing the intelligent information retrieval program is downloaded into the structured document browsing means 3a. Then, the structured document browsing means 3a executes the intelligent information retrieval program contained in the downloaded document. As a result, questions defined by the table defining means 1a and answer alternatives to the questions are displayed on a display screen of the client 3. When the user inputs answers, the inputted answers and the knowledge established by the knowledge establishing means 1b are compared with each other, producing a conclusion. If the conclusion is represented by the address of a document within a database 2b on the server 2, then the document is downloaded and browsed by the structured document browsing means 3a.

The intelligent information retrieval program gener-

ation system 1 allows a document containing an intelligent information retrieval program to be easily generated in a tabular form. If the document containing the intelligent information retrieval program is used by a WWW server on the Internet, then the burden on the WWW server can be reduced because an information retrieval process is carried out at a WWW terminal. An embodiment of the present invention which is applied to a WWW server on the Internet will be described below.

FIG. 2 shows in block form a network system to which the present invention is applied. In FIG. 2, a WWW server 10 and a WWW terminal 20 are connected to each other through the Internet 30.

The WWW server 10 has an intelligent information retrieval program generator 11. The intelligent information retrieval program generator 11 includes a table defining unit 11a for receiving questions to be used as criteria for determining conclusions, answer alternatives to the questions, and information relative to conclusions. The conclusions may represent document information to be provided to the user or a table in which other knowledge has been inputted. The table defining unit 11a generates a table 11b from inputted information. The intelligent information retrieval program generator 11 also includes a knowledge establishing unit 11c for receiving knowledge which is inputted by the operator as true or false with respect to each of the answer alternatives. The knowledge establishing unit 11c establishes the inputted knowledge as "O" or "X" with respect to each of the answer alternatives in the table 11b. The intelligent information retrieval program generator 11 further includes a code generator 11d for generating an intelligent information retrieval program based on the table 11b in which the knowledge has been established. The intelligent information retrieval program is generated using Javascript (the script language jointly developed by Sun Microsystems, Inc. in U.S.A. and Netscape Communications Corp. in U.S.A.) and Java applets (the software parts generated by the object-oriented language "Java" developed for the Internet by Sun Microsystems, Inc. in U.S.A.). The code generator 11d incorporates the generated intelligent information retrieval program into a HTML document, thereby producing a retrieval-program-containing document 14. The WWW server 10 has a server software program 12 which sends information contained in a home page 13, the retrieval-program-containing document 14, and a database 15 to the WWW terminal 20. The information in the database 15 includes document information, image information, and audio information, etc.

The WWW terminal 20 has a WWW browser 21. The WWW browser 21 has a function to browse HTML documents and a function to execute Java scripts and Java applets. Java applets are generated as intermediate codes independent of the OS (Operating System) and the MPU (Micro Processing Unit), enabling the interpreter of the WWW browser 21 to absorb differences between OSs and MPUs. Programs such as Java

scripts and Java applets that do not depend on the OS and the MPU will hereinafter be referred to as an "intermediate language".

The provider of information at the WWW server 10 generates information to be provided and stores the generated information in the database 15. The provider then generates the retrieval-program-containing document 14 in a HTML format using the functions of the intelligent information retrieval program generator 11. The user downloads the retrieval-program-containing document 14 by selecting certain items from a menu in the home page 13. When the WWW browser 21 executes the retrieval-program-containing document 14, questions and answer alternatives are displayed on a display screen of the WWW terminal 20. The user then inputs answers to the questions. When the user inputs answers to all the questions, information depending on the entered answers is acquired from the database 15, and displayed on the display screen.

A process of generating an intelligent information retrieval program, which is carried out by the intelligent information retrieval program generator 11, will be described below. In the process described below, an intelligent information retrieval program is generated using Java scripts.

FIG. 3 is a flowchart showing such a process of generating an intelligent information retrieval program. In the flowchart shown in FIG. 3 and other flowcharts described later on, numerals preceded by a prefix "S" represent step numbers.

[S1] The table defining unit 11a defines a table 11b of answers to questions and conclusions. Details of this process will be described later on with reference to FIG. 4.

[S2] The knowledge establishing unit 11c inputs knowledge into the table 11b defined in the step S1. Details of this process will be described later on with reference to FIG. 5.

[S3] The code generator 11d encodes the knowledge into Java scripts, and incorporates the Java scripts into a HTML document. Details of this process will be described later on with reference to FIG. 6.

FIG. 4 is a flowchart showing details of the process of defining the table 11b. The process of defining the table 11b is carried out by the table defining unit 11a based on data inputted by the operator.

[S11] The title of the table which is about to be defined is determined.

[S12] Questions in the table are determined.

[S13] Answer alternatives to each of the questions are determined.

[S14] Conclusions or another table which is a transition from the table is determined. In this manner, the table 11b is defined.

FIG. 5 is a flowchart showing details of the process of inputting knowledge into the table 11b.

[S21] The defined table is opened.

[S22] "O" or "X" is applied to each of the answer alternatives corresponding to the transition to the conclusion in each row or another table. In this manner, the correspondence between a combination of "O" or "X" for all the answer alternatives and the conclusions or another intelligent information retrieval program is determined.

FIG. 6 is a flowchart showing details of the process of generating codes.

[S31] A question display code and an answer alternative input code are generated.

[S32] A set of questions is represented by $Q = \{q_1, q_2, \dots, q_i\}$.

[S33] A first element of the set Q is represented by q.

[S34] A set of answer alternatives to the first element q is represented by A_q .

[S35] It is determined whether a is selected or not with respect to all elements a of the set A_q , thereby generating a code for a conditional branch.

[S36] It is determined whether the number of elements of the set Q is 1 or not. If the number of elements is 1, then control proceeds to a step S38. If the number of elements is not 1, then control proceeds to a step S37.

[S37] The first element of the set Q is removed from the set Q, and control returns to the step S33. In this fashion, the steps S33 through S37 are recursively executed with respect to all the elements of the set Q.

[S38] The signs "O" in the table are referred to, generating a code for a transition to a conclusion or another table corresponding to the combination of the present answers. In this manner, a retrieval program corresponding to each table is generated.

If a table is defined and knowledge is inputted through a menu, then such a process is highly convenient as it is free of any input oversights. An example of generating a retrieval-program-containing document using a knowledge editor by defining a table and inputting knowledge through a menu will be described below. In the following example, an intelligent information retrieval program for providing information as to banquet halls (restaurants) to a user who is looking for one

FIG. 7 shows a displayed image of a knowledge editor. When the knowledge editor is started, a main window 41 is displayed. The main window 41 has a menu bar 41a in which the menu name "File" is displayed. When "File" is selected, a pull-down menu 41b is opened therebelow. The pull-down menu 41b contains commands "Save as", "Print", and "Quit". If "Save as" is

executed, then knowledge information which is currently generated is saved under an arbitrary file name. If "Print" is executed, then knowledge information is printed. If "Quit" is executed, then the knowledge editor is finished.

The main window 41 contains a display area 41c for displaying a list of questions (Q list). The display area 41c displays the contents of questions inputted in a Q editing window (described later on) together with question numbers. To the right of the display area 41c, there is displayed a corresponding relationship between question numbers (Q1, Q2) and matrix numbers. The matrix numbers are serial numbers allotted to defined tables.

The main window 41 also contains, in its central area, a matrix table 41d of answer alternatives and conclusions. Above the matrix table 41d, there are displayed question numbers (Q1, Q2) of the defined questions, numbers (1:1, 1:2, 2:1, 2:2) indicative of alternatives for each of the questions, and answer numbers (A1, A2). To the left of the matrix table 41d, there are displayed conclusion numbers (R1, R2, R3) and sequence numbers (010, 020, 030) of the conclusions. The matrix table 41d contains true "O" and false "X" with respect to each of the answer alternatives for reaching conclusions.

The main window 41 further includes, displayed below the matrix table 41d, a "Q edit" button 41e for inputting questions, an "R edit" button 41f for inputting conclusions, and a "sequence No. change" button 41g for changing sequence numbers. When the "Q edit" button 41e is clicked with a mouse, a Q editing window is newly displayed. When the "R edit" button 41f is clicked with the mouse, an R editing window is newly displayed. An A editing window can further be displayed from the Q editing window. When the "sequence No. change" button 41g is clicked with the mouse, the sequence numbers applied to the conclusions can be changed, and the conclusions are rearranged in the order of the changed sequence numbers.

FIG. 8 shows the Q editing window. The Q editing window, which is denoted by 42 in FIG. 8, has a question inputting area 42a which contains input items "Question No.", "Sequence No.", "Brief name", "Displayed contents", "Reference information", "Reference pattern", "URL", and "A edit".

The input item "Question No." represents an identification number assigned to a registered question. The input item "Sequence No." represents a sequence number assigned to a question. When a new question is inputted, it is automatically given a sequence number in tens. The input item "Brief name" represents an abbreviated name given to a question to make the content of the question readily understandable. The input item "Displayed contents" represents a specific content of a question. The input item "Reference information" represents whether there is reference information or not as indicated by a check box. The input item "Reference pattern" represents an identification number showing the

type of a reference pattern. There are three types of patterns available. According to the first pattern (identification number "1"), alternatives are displayed by list boxes. According to the second pattern (identification number "2"), alternatives are displayed by image data. According to the third pattern (identification number "3"), another window is displayed for selecting an answer. The input item "URL" represents an area for inputting a URL of another document if a reference pattern exists as the other document. The input item "A edit" includes a button for opening the A editing window for establishing an answer to a question. If an A editing process has already been carried out, then a check mark is displayed in the button.

The Q editing window 42 also contains an array of "Q rearrange", "Add", "Delete", "Save", and "Cancel" buttons 42b ~ 42f in a lower area thereof. When the "Q rearrange" button 42b is clicked with the mouse, questions are rearranged into the sequence of the "Sequence No." at the time. If the sequence of questions is to be changed, then the "Sequence No." is changed into a desired number, and the "Q rearrange" button 42b is clicked with the mouse. When the "Add" button 42c is clicked with the mouse, a new question is added to a position which is being indicated at the time. When the "Delete" button 42d is clicked with the mouse, the question which is being indicated at the time is deleted. When the "Save" button 42e is clicked with the mouse, the data established on the Q editing window 42 are saved, and the content thereof is reflected in the matrix table 41d in the main window 41. When the "Cancel" button 42f is clicked with the mouse, all the data established on the Q editing window 42 are cleared.

FIG. 9 shows the A editing window. The A editing window, which is denoted by 43 in FIG. 9, contains "Question No." of a corresponding question displayed in an upper area thereof and also an answer inputting area 43a displayed in a central area thereof. The answer inputting area 43a contains input items "Answer No.", "Sequence No.", "Brief name", and "Displayed contents". These input items are identical to the input items "Answer No.", "Sequence No.", "Brief name", and "Displayed contents" in the question inputting area 42a of the Q editing window 42.

The A editing window 43 also contains an array of "A rearrange", "Register", "Delete", and "Cancel" buttons 43b ~ 43e in a lower area thereof. The functions of the "A rearrange", "Delete", and "Cancel" buttons 43b, 43d, 43e are identical respectively to the functions of the "Q rearrange", "Delete", and "Cancel" buttons 42b, 42d, 42f in the Q editing window 42. When the "Register" button 43c is clicked with the mouse, the data established in the A editing window 43 are registered as an answer to the corresponding question.

FIG. 10 shows the R editing window. The R editing window, which is denoted by 44 in FIG. 10, contains conclusion inputting area 44a displayed in a central area thereof. The conclusion inputting area 44a contains in-

put items "Conclusion No.", "Sequence No.", "Brief name", "Conclusion or Related table" (displayed as "C/R" in the window), and "URL". The input items "Conclusion No.", "Sequence No.", and "Brief name" are identical to the input items "Answer No.", "Sequence No.", and "Brief name" in the question inputting area 42a of the Q editing window 42. The input item "Conclusion or Related table" has two check boxes, one representing the immediate display of a conclusion and the other representing a further interactive process using another related table. The input item "URL" represents an area for inputting a URL of information of a conclusion or an URL of a related table.

The R editing window 44 also contains an array of "R rearrange", "Save", "Delete", and "Cancel" buttons 44b ~ 44e in a lower area thereof. The functions of the "R rearrange", "Save", "Delete", and "Cancel" buttons 44b ~ 44e are identical respectively to the functions of the "Q rearrange", "Save", "Delete", and "Cancel" buttons 42b, 42e, 42d, 42f in the Q editing window 42.

When necessary items are inputted and saved in the Q editing window 42, the A editing window 43, and the R editing window 44, a Q table, an A table, and an R table are generated in the intelligent information retrieval program generation system 1, and a matrix table is generated from these Q, A, and R tables in the intelligent information retrieval program generation system 1.

FIG. 11 shows the Q, A, and R tables and the matrix table which are generated in the intelligent information retrieval program generation system 1. The Q table, denoted by 51 in FIG. 11, contains items "Q No.", "Q brief name", "Q displayed name", "Reference pattern", and "QURL" registered therein. The A table, denoted by 52 in FIG. 11, contains items "Q No.", "A No.", "Sequence No.", "A brief name", and "A display name" registered therein. The R table, denoted by 53 in FIG. 11, contains items "R No.", "Sequence No.", "R brief name", "RURL", and "R related table" registered therein.

The matrix table, denoted by 54 in FIG. 11 is generated from the information contained in the Q, A, and R tables 51, 52, 53. The matrix table 54 represents a corresponding relationship between answers and conclusions. The conclusions are given "Sequence No.". Each of the answers is assigned a two-figure number, the tenth digit representing "Question No." and the unit digit representing "Answer No.". The matrix table 54 is displayed as the matrix table 41d in the main window 41 shown in FIG. 7. When knowledge is inputted as "O" or "X" in the main window 41, a corresponding value is established in the matrix table 54. Specifically, when knowledge is inputted as "O", a corresponding value "1" is established in the matrix table 54, and when knowledge is inputted as "X", a corresponding value "0" is established in the matrix table 54. In this manner, the corresponding relationship between the entered knowledge and the conclusions or another table is determined. The matrix table 54 in which knowledge has

been inputted will hereinafter be referred to as tabular knowledge data.

FIG. 12 is a flowchart of a process of generating an intelligent information retrieval program with the knowledge editor. The process shown in FIG. 12 is carried out by the operator while the knowledge editor is providing an operating environment for the operator and storing and analyzing various items of information that have been entered.

[S41] The operator inputs a question. If, at this time, the Q editing window 42 is closed, then the operator presses the "Q edit" button 41e to open the Q editing window 42, and inputs a question. Specifically, the operator inputs "Question No.", "Sequence No.", "Brief name", "Displayed contents", "Reference information", and "Reference pattern". If necessary, the operator also inputs "URL" of the reference information. The knowledge editor stores information relative to the entered question.

[S42] The operator determines whether all questions have been inputted or not. If there is a question which has not been inputted, control goes back to the step S41. If all questions have been inputted, then control goes to a step S43.

[S43] The operator inputs an answer. If, at this time, the A editing window 43 is closed, then the operator presses an "A edit" button to open the A editing window 43, and inputs an answer. Specifically, the operator inputs "Answer No.", "Sequence No.", "Brief name", and "Displayed contents". The knowledge editor stores information relative to the entered answer.

[S44] The operator determines whether all answers have been inputted or not. If there is an answer which has not been inputted, control goes back to the step S43. If all answers have been inputted, then control goes to a step S45.

[S45] The operator inputs a conclusion. If, at this time, the R editing window 44 is closed, then the operator presses the "R edit" button 41f to open the R editing window 44, and inputs a conclusion. Specifically, the operator inputs "Conclusion No.", "Sequence No.", "Brief name", "Conclusion or Related table", and "URL". The knowledge editor stores information relative to the entered conclusion.

[S46] The operator determines whether all conclusions have been inputted or not. If there is an answer which has not been inputted, control goes back to the step S45. If all answers have been inputted, then control goes to a step S47. At this time, the matrix table 41d has been generated in the main window 41 of the knowledge editor.

[S47] The operator inputs knowledge. Specifically, the operator enters "O" and "X" in the matrix table 41d which has been generated in the above process up to the step S46.

[S48] The knowledge editor then effects a unique-

ness check and a completeness check on the defined matrix table 41d and the knowledge. The uniqueness check is a checking process for determining whether one conclusion can be derived from an arbitrary pattern of answers. The completeness check is a checking pattern for determining whether information to be inputted is consistent or not.

[S49] The operator determines whether the tabular knowledge data are normal or not based on the results of the checks effected in the step S48. If the tabular knowledge data are inconsistent, then control goes back to the step S41, S43, or S45 depending on the judgment of the operator. If the tabular knowledge data are normal, then control proceeds to a step S50.

[S50] The knowledge editor converts the tabular knowledge data into the intermediate language (Java scripts and Java applets).

[S51] The knowledge editor converts related tabular knowledge data into the intermediate language. Specifically, if another table is indicated in a conclusion, the other table is converted into the intermediate language. The specific process carried out in each of the steps S50, S51 is the process of generating codes shown in FIG. 6.

FIG. 13 shows, by way of example, the generated tabular knowledge data. The tabular knowledge data, denoted by 11ba, which is referred to as "table1", contain two questions "Q1", "Q2" each having two answer alternatives "A1", "A2", and two conclusions "result1", "result2" and two other tables "table2", "table3" which are defined with respect to the two questions "Q1", "Q2". The tabular knowledge data, denoted by 11bb, which is referred to as "table2", contain a question "Q1" having two answer alternatives "A1", "A2", and two conclusions "result3", "result4" which are defined with respect to the question "Q1". The tabular knowledge data, denoted by 11bc, which is referred to as "table3", contain a question "Q1" having two answer alternatives "A1", "A2", and two conclusions "result5", "result6" which are defined with respect to the question "Q1".

Based on the tabular knowledge data, codes are generated to produce a HTML document which contains an intelligent information retrieval program generated using Java scripts.

FIG. 14 shows, by way of example, a document containing an intelligent information retrieval program that has been generated based on the tabular knowledge data referred to as "table1". In FIG. 14, an area 61 which is present between tags <SCRIPT>, </SCRIPT> is a description of a program generated using Java scripts. The program description contains conditional branches "if ... else" based on whether an answer alternative is selected or not. A code for such a conditional branch is generated in the step S35 shown in FIG. 6. A code indicated by "url = ..." shows a destination, represented by an URL, of a transition to a conclusion or a table. A code

for indicating the destination is generated in the step S38 shown in FIG. 6.

In FIG. 14, an area 62 which is present between tags <BODY>, </BODY> is a description of a code for displaying questions and answer alternatives and selectively receiving answers. This code is generated in the step S31 shown in FIG. 6.

FIG. 15 shows, by way of example, a document containing an intelligent information retrieval program that has been generated based on the tabular knowledge data referred to as "table2". In FIG. 15, an area 63 which is present between tags <SCRIPT>, </SCRIPT> is a description of a program generated using Java scripts, and an area 64 which is present between tags <BODY>, </BODY> is a description of a code for displaying questions and answer alternatives and selectively receiving answers.

FIG. 16 shows, by way of example, an HTML document of a conclusion. The displayed content of a conclusion is described in an area 65 which is present between tags <BODY>, </BODY>. The example shown in FIG. 16 illustrates the displayed content of "Conclusion 5". Other conclusions can be generated similarly. The conclusions may contain image data and audio data.

FIG. 17 is a flowchart showing a knowledge structure of the intelligent information retrieval program that has been generated based on the tabular knowledge data referred to as "table1".

[S61] It is determined whether the answer to the question Q1 in "table1" is A1 or not. If the answer is A1, then control goes to a step S62, and if the answer is not A1, then control goes to a step S63.

[S62] It is determined whether the answer to the question Q2 in "table1" is A1 or not. If the answer is A1, then "result1" is used as the conclusion, and if the answer is not A1, then control goes to a retrieval program generated based on "table2".

[S63] It is determined whether the answer to the question Q2 in "table1" is A1 or not. If the answer is A1, then "result2" is used as the conclusion, and if the answer is not A1, then control goes to a retrieval program generated based on "table3".

FIG. 18 is a flowchart showing a knowledge structure of the intelligent information retrieval program that has been generated based on the tabular knowledge data referred to as "table2".

[S64] It is determined whether the answer to the question Q1 in "table2" is A1 or not. If the answer is A1, then "result3" is used as the conclusion, and if the answer is not A1, then "result4" is used as the conclusion.

FIG. 19 is a flowchart showing a knowledge structure of the intelligent information retrieval program that has been generated based on the tabular knowledge

data referred to as "table3".

[S65] It is determined whether the answer to the question Q1 in "table3" is A1 or not. If the answer is A1, then "result5" is used as the conclusion, and if the answer is not A1, then "result6" is used as the conclusion.

In the manner described above, an HTML document which contains an intelligent information retrieval program generated using Java scripts is generated by the knowledge editor. The home page 13 provided by the WWW server 10 and the generated HTML document are related to each other for thereby providing a user who has accessed the home page 13 with an intelligent information retrieval capability. Examples of displayed images at the WWW terminal 20 which are provided with services of the generated intelligent information retrieval program will be described below.

FIG. 20 shows, by way of example, a transition from the home page to the document containing the intelligent information retrieval program. When the user accesses the home page 13 using the WWW browser 21, a top menu 71 is displayed on the display screen of the WWW terminal 20. In the example shown in FIG. 20, the displayed menu 71 contains items "1. Looking for a banquet hall", "2. Date course guide", "3. Shopping guide", etc. When the item "1. Looking for a banquet hall" is selected, the document containing the intelligent information retrieval program is downloaded, and an image 72 of questions and answers to look for a banquet hall is displayed on the display screen.

The displayed image 72 contains "Question 1" and "Question 2". The user first selects an answer to "Question 1: How much is the budget per person?". Specifically, the user selects an answer from a group of answers displayed in a list box 72a positioned to the right of the displayed content of "Question 1". In this example, the list box 72a contains two answers, i.e., "Less than 5000 yen" and "At least 5000 yen".

Then, the user selects an answer to "Question 2: What kind of food do you like?" FIG. 21 shows how an answer is made to "Question 2". Below the displayed content of "Question 2", there are displayed answer alternatives each comprising image data such as a photograph and a food description. The image data can be displayed on an enlarged scale. In this example, the answer alternatives include an answer 72b for Chinese food and an answer 72c for Western food. The answers 72b, 72c are associated with respective check boxes, and the user checks one of the check boxes for the food which the user likes.

If Western food is chosen, then a selection image 73 containing "Question 3: Do you like French food or Italian food?" is displayed for the user to select a type of Western food. FIG. 22 shows how an answer is made to "Question 3". The selection image 73 contains an answer button 73a displayed below the displayed content

of "Question 3". When the user presses the answer button 73a, an answer selection image 74 is displayed on the display screen. The answer selection image 74 contains answer alternatives each comprising image data such as a photograph and a food description. When the user selects an answer in the answer selection image 74, the answer selection image 74 changes back to the selection image 73.

When the user then presses a guide button 73b in the selection image 73, introduction information relative to a restaurant as a banquet hall is displayed as a conclusion on the basis of the answers which the user has inputted to the three questions described above. FIG. 23 shows, by way of example, a displayed image 76 of such a conclusion. In the example shown in FIG. 23, the displayed image 76 contains the image of an interior of the restaurant, a map showing the location of the restaurant, and detailed information of the restaurant. The detailed information includes the name of the restaurant, the style of cooking, business hours, telephone number, the price of package food, regular holiday, and special benefits.

The displayed image 76 also contains a button 76a for returning to the top menu 71 below the introduction information relating to the restaurant. Pressing the button 76a immediately brings the user back to the top menu 71.

If answers are incomplete and no conclusion can be formed, an announcement that desired information cannot be displayed is displayed. FIG. 24 shows a displayed image 77 of an announcement which is made when answers are incomplete. The displayed image 77 contains a message indicating that no desired information can be displayed and a reason why no desired information can be displayed. The displayed image 77 contains in its lower area a button 77a for going to a questionnaire image and a button 77b for going to a top menu. When the button 77a is pressed, the displayed image 77 changes back to the image 72 of questions and answers. When the button 77b is pressed, the displayed image 77 changes back to the top menu 71.

In the manner described above, an intelligent information retrieval program can easily be generated in a tabular form. Since the generated intelligent information retrieval program is incorporated in an HTML document, it can be downloaded through the Internet to a WWW terminal, and desired information can be retrieved on an interactive basis at the WWW terminal using the intelligent information retrieval program.

In the above embodiment, it has been described that an intelligent information retrieval program is generated using Java scripts. However, an intelligent information retrieval program may be generated with Java. If an intelligent information retrieval program is generated with Java, then it is necessary to compile source codes into Java applets. To incorporate Java applets into an HTML document, the names of the Java applets, the sizes of the Java applets, and arguments to be

passed to the Java applets may be indicated in the HTML document.

While only one conclusion is drawn from answers to a set of questions in the illustrated embodiment, a plurality of conclusions may be drawn from the answers. According to such a modifications, a plurality of conclusions are simultaneously displayed using a frame division function of a WWW browser.

In the intelligent information retrieval program generation system according to the present invention, as described above, when answer conditions are established in a conclusion determining table of answer alternatives to questions and conclusions, there is generated a document which incorporates an intelligent information retrieval program that effects processing based on the conclusion determining table. Therefore, it is possible to easily generate an intelligent information retrieval program which is applicable to a wide-area information network.

In the intelligent information retrieval system according to the present invention, a document having a logic structure according to a predetermined information description language and incorporating an intelligent information retrieval program is stored in a retrieval-program-containing document storage means, and the document is browsed and the intelligent information retrieval program is executed by a structured document browsing means. Consequently, intelligent information can be retrieved at a terminal where such intelligent information should be received.

The forgoing is considered as illustrative only for the principles of the present invention. Further, since numerous modifications and changes will readily occur to those skilled in the art, it is not desired to limit the invention to the exact construction and applications shown and described, and accordingly, all suitable modifications and equivalents may be regarded as falling within the scope of the invention in the appended claims and their equivalents.

Thus, in summary, the present invention can provide a system which can easily generate an intelligent information retrieval program that is applicable to a wide-area information network such as the Internet. When questions to be used as criteria for determining conclusions, answer alternatives to the questions, and information of conclusions are inputted, a table defining unit defines a conclusion determining table which comprises answer alternatives to each of the questions and conclusions. A knowledge establishing unit displays the conclusion determining table on a display screen, receives inputted answer conditions for the conclusions to be selected, and establishes the answer conditions in the conclusion determining table. A retrieval program generator generates an intelligent information retrieval program for decoding the conclusion determining table in which the knowledge has been established, displaying questions, receiving answers to the questions, and determining conclusions depending on the answers.

The retrieval program generator further generates a document containing the intelligent information retrieval program by incorporating the generated intelligent information retrieval program into a document description in the form of hypertext.

Claims

1. An intelligent information retrieval program generation system for assisting in generating a program for providing an intelligent information retrieval function, comprising:

table defining means for receiving questions to be used as criteria for determining conclusions, answer alternatives to the questions, and information of conclusions, and defining a conclusion determining table which comprises said answer alternatives and said conclusions; knowledge establishing means for displaying said conclusion determining table on a display screen, receiving inputted answer conditions for said conclusions to be selected, and establishing said answer conditions in said conclusion determining table; and retrieval program generating means for generating, based on said conclusion determining table, an intelligent information retrieval program for receiving answers inputted to said questions on an interactive basis and displaying conclusions in response to the answers, and incorporating said intelligent information retrieval program into a document having a logic structure according to a predetermined information description language.

2. An intelligent information retrieval program generation system according to claim 1, wherein said table defining means comprises means for indicating, as said information of conclusions, a document incorporating another intelligent information retrieval program generated based on another conclusion determining table.

3. An intelligent information retrieval program generation system according to claim 1 or 2, wherein said retrieval program generating means comprises means for incorporating said intelligent information retrieval program into a hypertext document.

4. An intelligent information retrieval program generation system according to claim 1, 2, or 3, wherein said retrieval program generating means comprises means for generating said intelligent information retrieval program with a predetermined script language.

5. An intelligent information retrieval program generation system according to claim 1, 2, 3, or 4, wherein said retrieval program generating means comprises means for generating said intelligent information retrieval program as software parts executable by a predetermined interpreter.

6. An intelligent information retrieval system for retrieving intelligent information on an interactive basis, comprising:

retrieval-program-containing document storage means for storing a retrieval-program-containing document incorporating an intelligent information retrieval program which has a logic structure according to a predetermined information description language for receiving answers inputted to questions on an interactive basis and displaying conclusions in response to the answers; and structured document browsing means for browsing a document having a logic structure according to said predetermined information description language, and browsing said retrieval-program-containing document and executing the intelligent information retrieval program incorporated in said retrieval-program-containing document when said retrieval-program-containing document is acquired from said retrieval-program-containing document storage means.

7. An intelligent information retrieval system according to claim 6, wherein said structured document browsing means comprises a World Wide Web Browser.

8. A method of generating a program for providing an intelligent information retrieval function, comprising steps of:

receiving questions to be used as criteria for determining conclusions, answer alternatives to the questions, and information of conclusions, and defining a conclusion determining table which comprises said answer alternatives and said conclusions; displaying said conclusion determining table on a display screen, receiving inputted answer conditions for said conclusions to be selected, and establishing said answer conditions in said conclusion determining table; and generating, based on said conclusion determining table, and intelligent information retrieval program for receiving answers inputted to said questions on an interactive basis and displaying conclusions in response to the answers, and incorporating said intelligent infor-

mation retrieval program into a document having a logic structure according to a predetermined information description language.

5

10

15

20

25

30

35

40

45

50

55

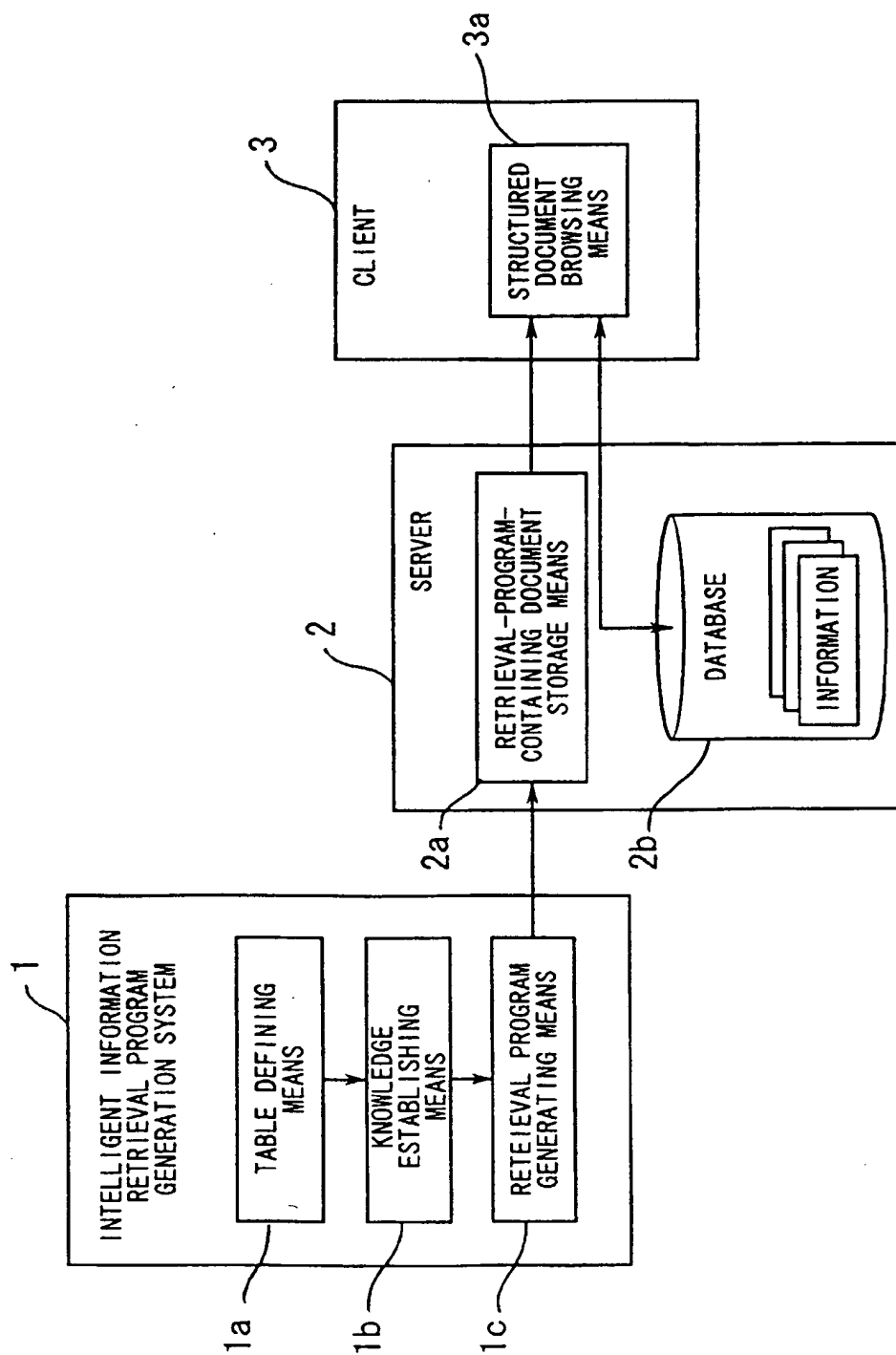


FIG. 1

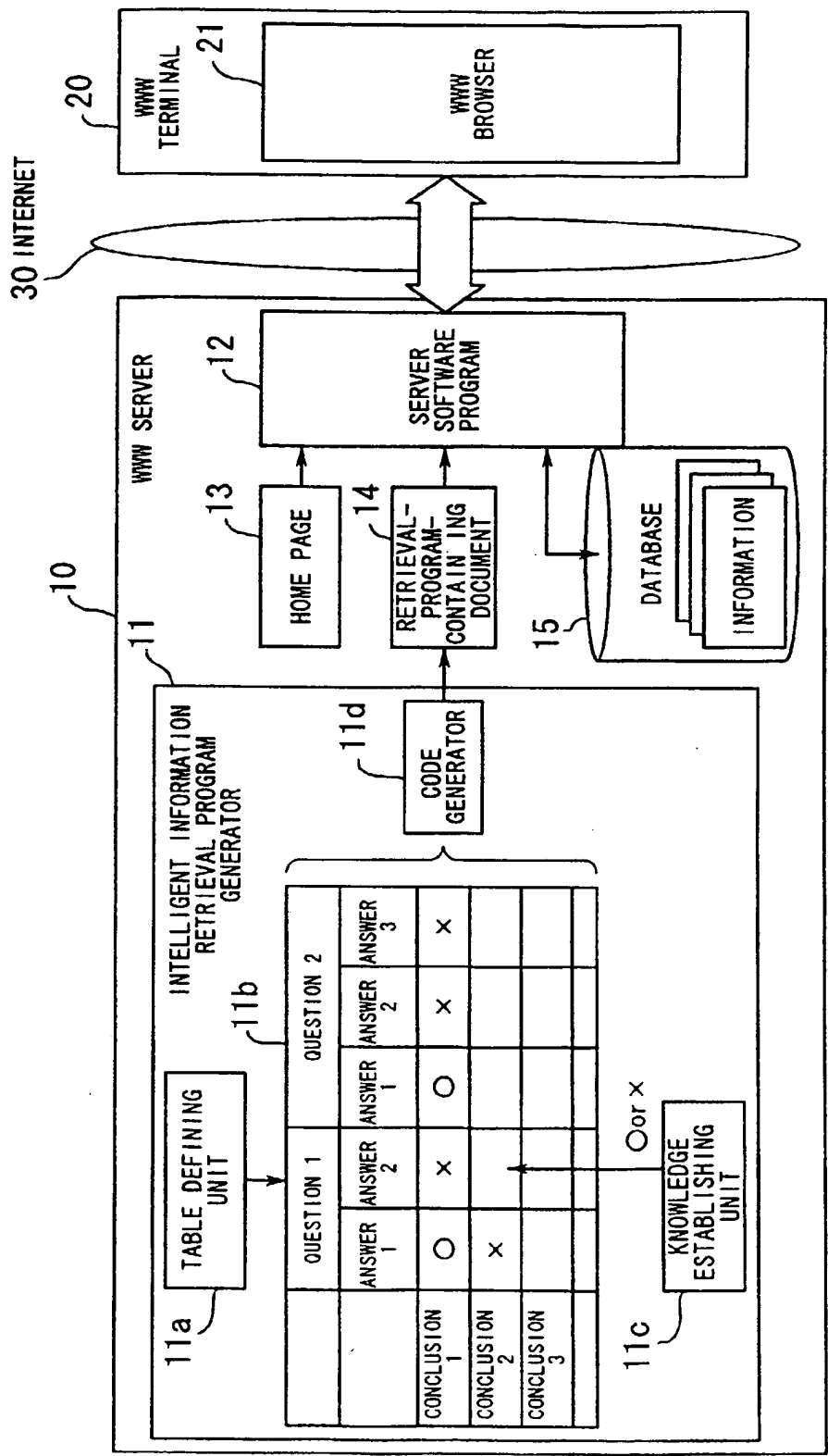


FIG. 2

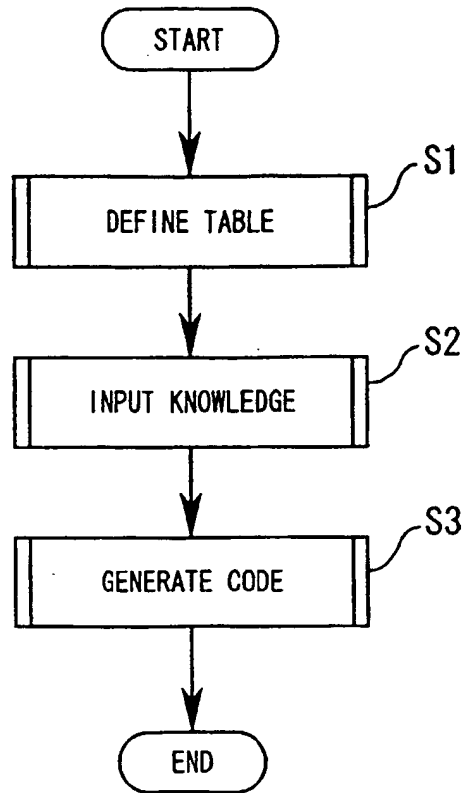


FIG. 3

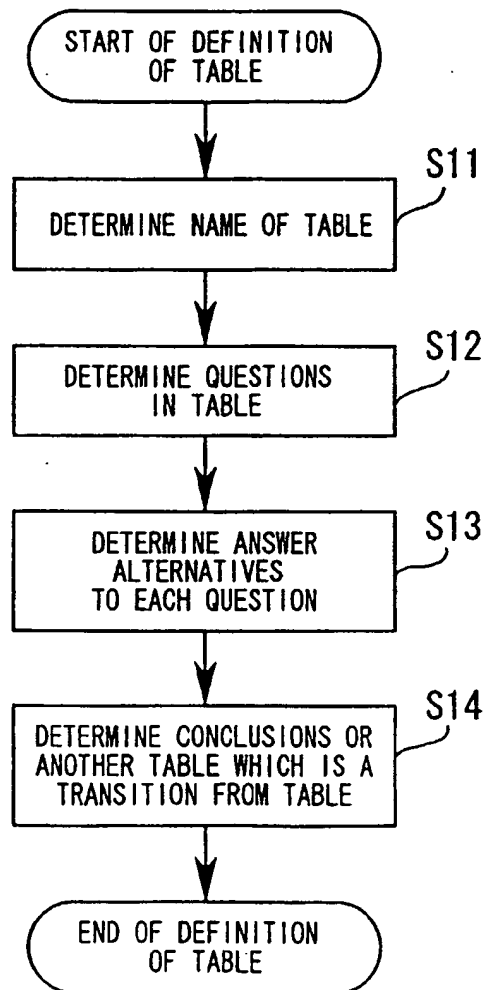


FIG. 4

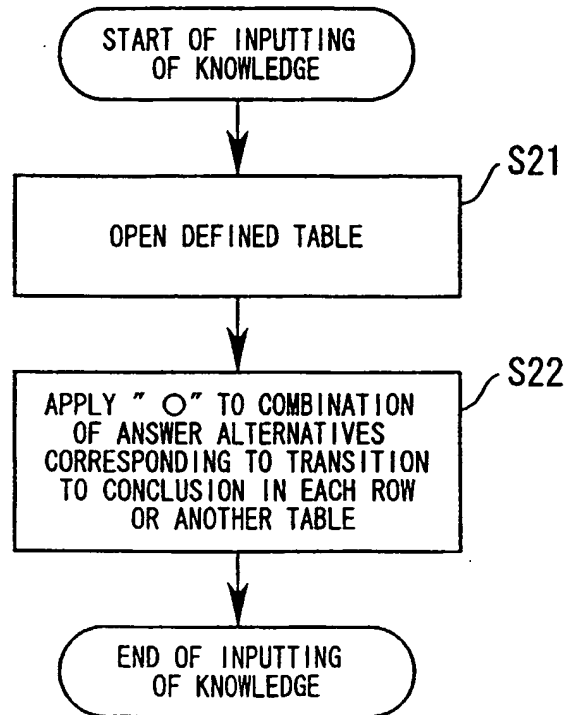


FIG. 5

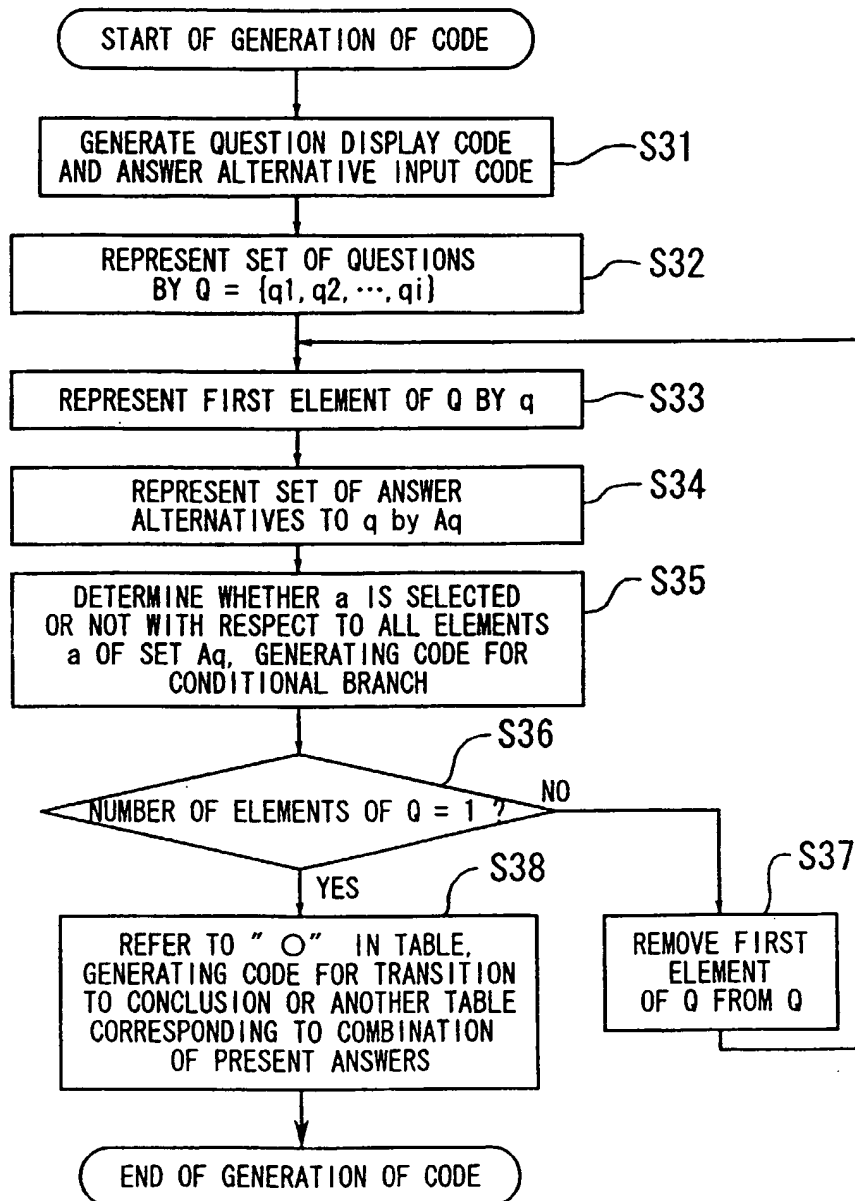


FIG. 6

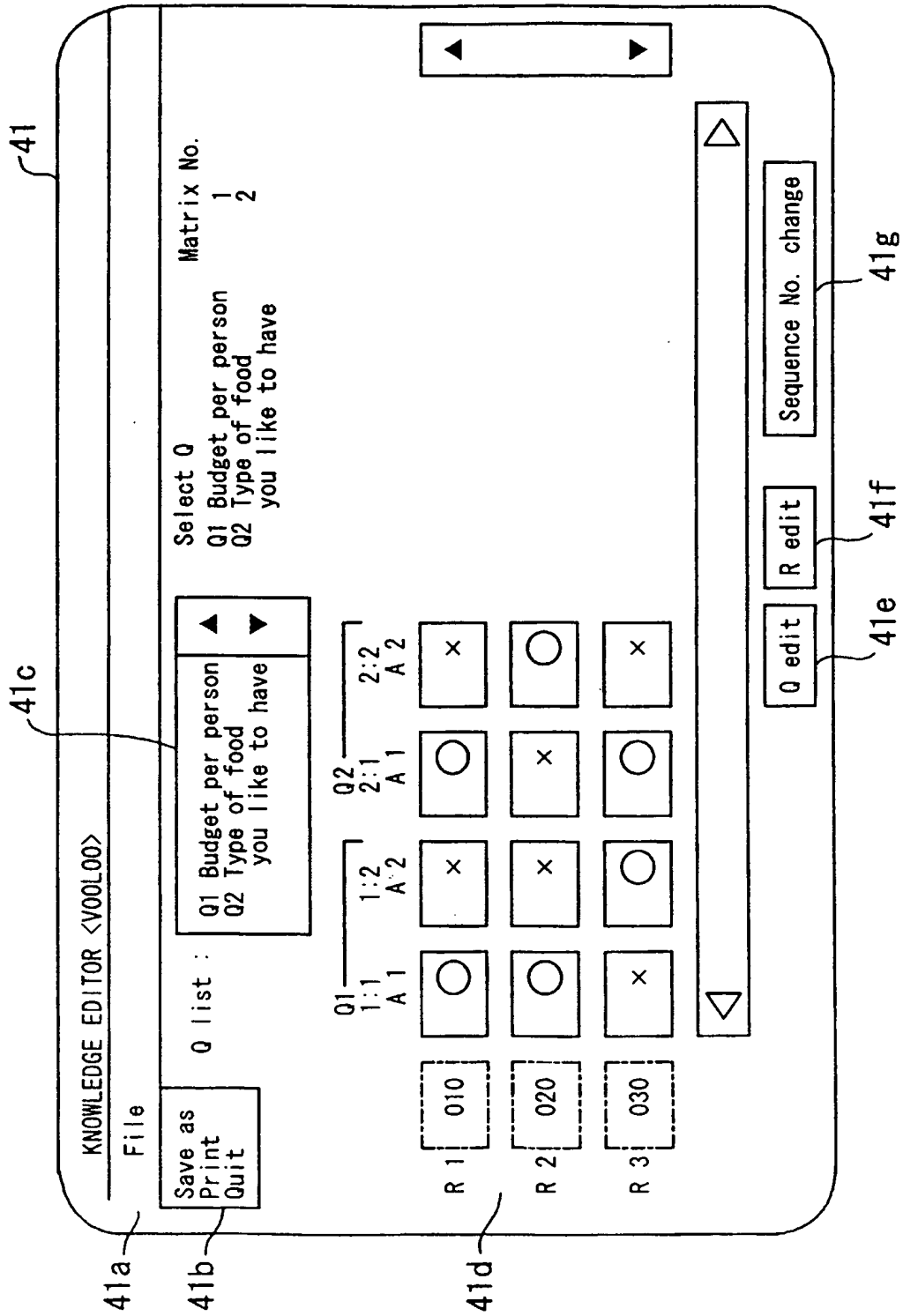


FIG. 7

42

42a

<0 edit>

Question No.	Sequence No.	Brief name	Displayed contents	Reference information	Reference pattern	U R L	A edit
1	10	Budget	Budget per person	<input type="radio"/> YES <input type="radio"/> NO	1		<input type="checkbox"/>
2	20	Food	Type of food you like to have	<input type="radio"/> YES <input type="radio"/> NO	2		<input type="checkbox"/>
.
.

Q rearrange

AddDeleteSaveCancel

42b42c42d42e42f

FIG. 8

43

43a

<A edit>

Q n O O O O

Answer No.	Sequence No.	Brief name	Displayed contents
1	10	Less than 5	Less than 5,000 yen
2	20	At least 5	At least 5,000 yen
:	:	:	:

43b

A rearrange

Register

Delete

Cancel

43c

43d

43e

FIG. 9

44

44a

<R edit>

Conclusion No.	Sequence No.	Brief name	C/R	URL
1	10	Japanese food 1	●/○	Conclusion1.html
2	20	TB2	○/●	Related2.html
:				
		R rearrange	Save	Delete
			Cancel	Cancel

44b 44c 44d 44e

FIG. 10

Q table

Q No.	Q brief name	Q displayed name	Reference pattern	Q U R L
1	Budget	Budget per person	1	
2	Food	Type of food you like to have	2	
3				

A table

Q No.	A No.	Sequence No.	A brief name	A displayed name
1	1	010	Less than 5	Less than 5,000 yen
1	2	020	Less than 5	At least 5,000 yen
2	1	030	Japanese food	Japanese food
2	2	010	Western food	Western food

R table

R No.	Sequence No.	R brief name	R U R L	R related table
1	010	Japanese food1	Conclusion1.html	
2	020	TB2		Related2.html
3	030	Japanese food2	Conclusion2.html	
4	040	TB3		Related3.html

Matrix table

	Sequence	1 1	1 2	2 1	2 2
R1	010	0	1	0	1
R2	020	0	1	1	0
R3	030	1	0	0	1
R4	040	0	1	1	0

FIG. 11

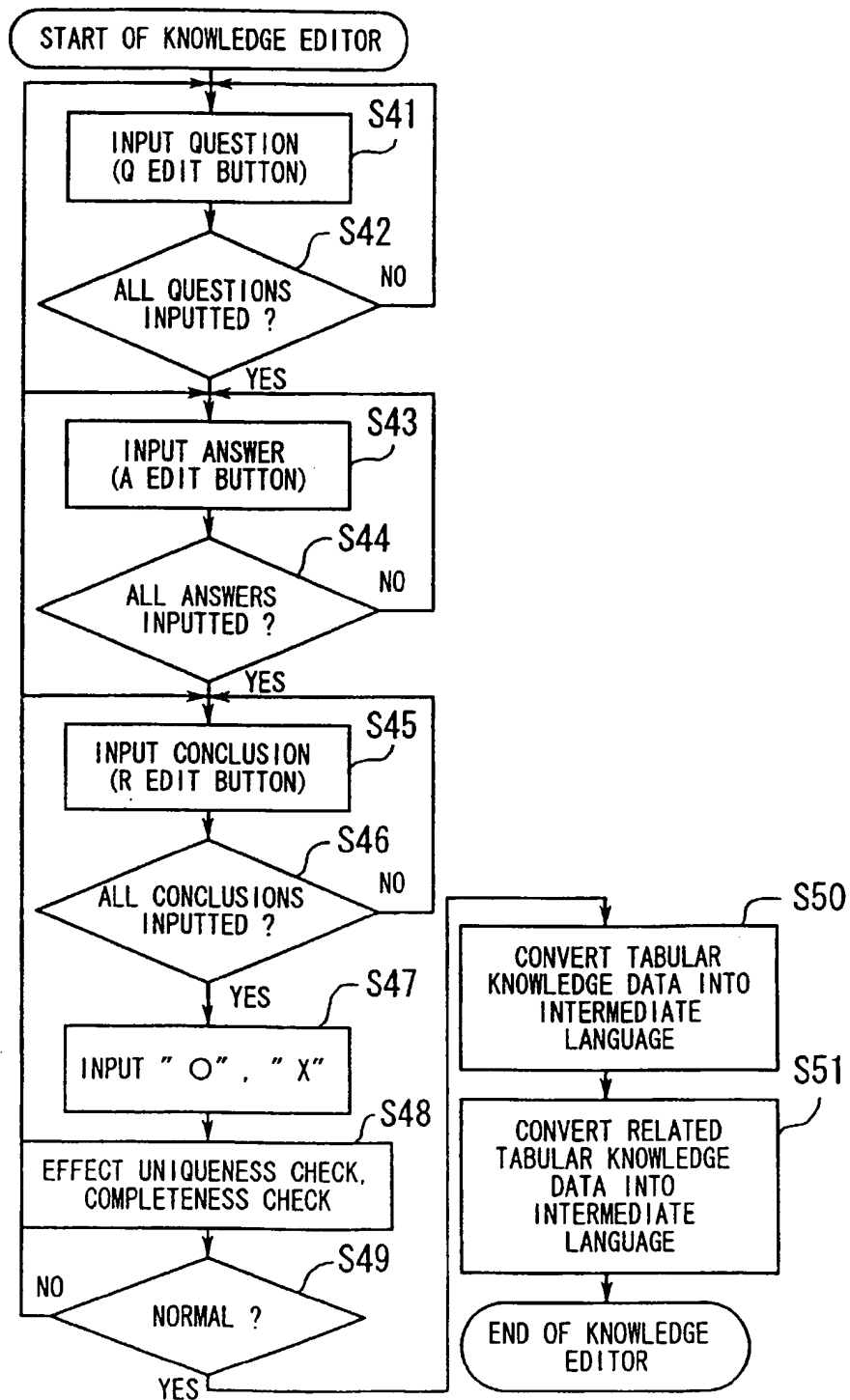


FIG. 12

11ba

table1	Q1		Q1	
	A1	A2	A1	A2
result1	○	×	○	×
table2	○	×	×	○
result2	×	○	○	×
table3	×	○	×	○

11bb

table2	Q1	
	A1	A2
result3	○	×
result4	×	○

11bc

table3	Q1	
	A1	A2
result5	○	×
result6	×	○

FIG. 13


```

<HTML>
<HEAD>
<TITLE>Table1</TITLE>
<SCRIPT>
function next_location (q1, q2 ) {
    var url
    if (q1.selectedIndex == 0) {
        if (q2[0].checked) {
            url = "result1.html"
        } else {
            url = "table2.html"
        }
    } else {
        if (q2[0].checked) {
            url = "result2.html"
        } else {
            url = "table3.html"
        }
    }
    window.location = url
    return true
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="questions">
<BR>
QUESTION 1: BUDGET PER PERSON ?
<SELECT NAME="Q1" SIZE="2">
<OPTION VALUE="A1" SELECTED <5.000 YEN
<OPTION VALUE="A2" ≥ 5.000 YEN
</SELECT>
<HR>
QUESTION 2: WHAT TYPE OF FOOD DO YOU LIKE TO HAVE ?
          (CHOOSE FROM PHOTOGRAPHS BELOW)

<BR>
<BR>
<TABLE BORDER>
<TR>
<TD><IMG SRC="picture.gif"></TD>
<TD><IMG SRC="picture.gif"></TD>
</TR>
<TR ALIGN=center>
<TC><INPUT TYPE="radio" NAME="Q2" CHECKED>CHINESE FOOD</TD>
<TC><INPUT TYPE="radio" NAME="Q2">WESTERN FOOD</TD>
</TR>
</TABLE>
<HR>
<INPUT TYPE="button" VALUE="OK" onClick="next_location
(document.questions.Q1, document.questions.Q2)">
</FORM>
</BODY>
</HTML>

```

61

62

FIG. 14

```

<HTML>
<HEAD>
<TITLE>Table2</TITLE>
<SCRIPT>
function openDialog (url) {
    dialogWin=window.open(url, "Dialog", "toolbar=no, scrollbar=no, width=250, height=180)
    dialogWin.from = window
    //dialogWin = window.open(url, "Dialog", "toolbar=no, scrollbar=no")
    return true
}
function next_location (q1_index) {
    var url
    if (q1_index == 0) {
        url = "result3.html"
    } else {
        url = "result4.thml"
    }
    window.location = url
    return true
}
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="questions">
<BR>3
QUESTION 3: FRENCH FOOD OR ITALIAN FOOD ?
<INPUT TYPE="button" VALUE="OK" onClick Me"onClick="openDialog('dialog0.html')">
<HR>
<INPUT TYPE="button" VALUE="OK" onClick= "next_location(q1_index)">
</FORM>
</BODY>
</HTML>

```

63

64

FIG. 15

```
<HTML>
<HEAD>
<TITLE>CONCLUSION5</TILTE>
</HEAD>
<BODY>
INTRODUCTION OF RESTAURANT: ITALIAN FOOD *** RESTAURANT } 65
</BODY>
</HTML>
```

FIG. 16

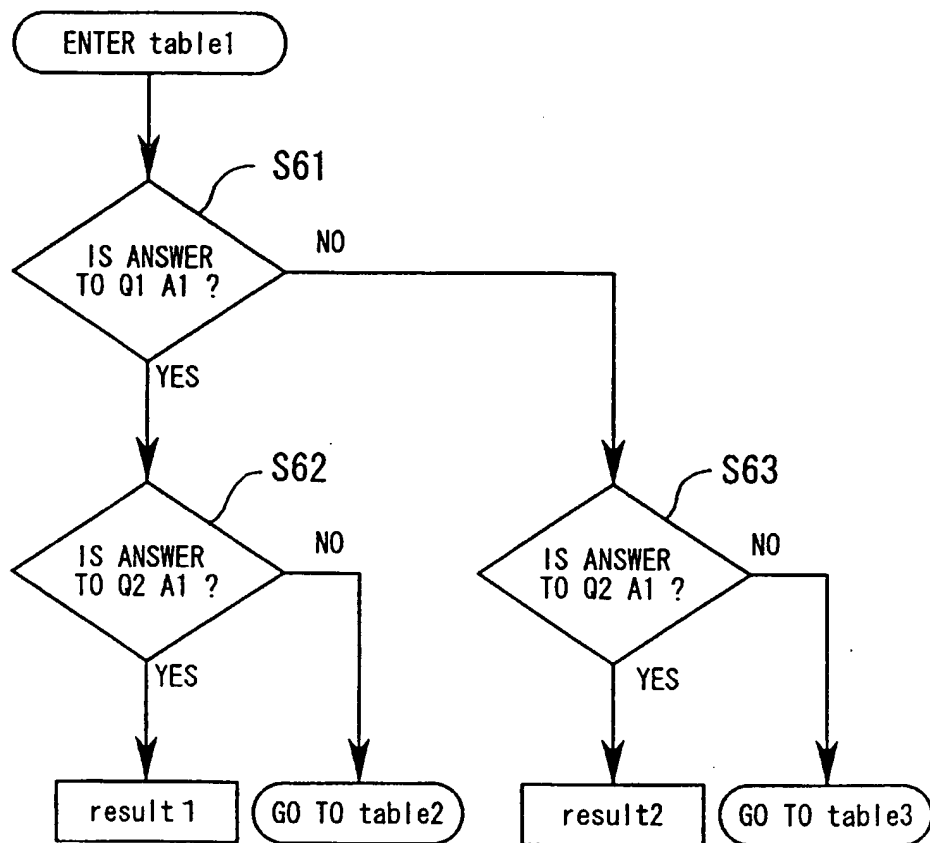


FIG.17

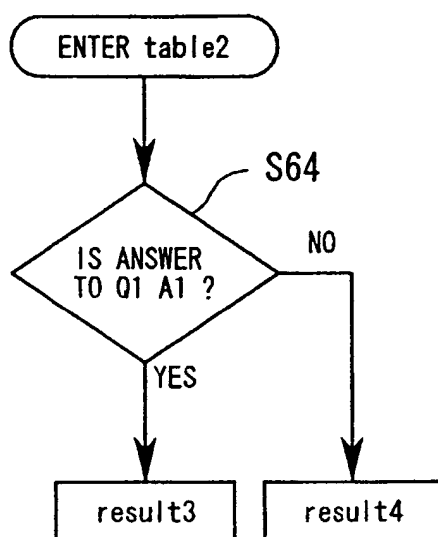


FIG. 18

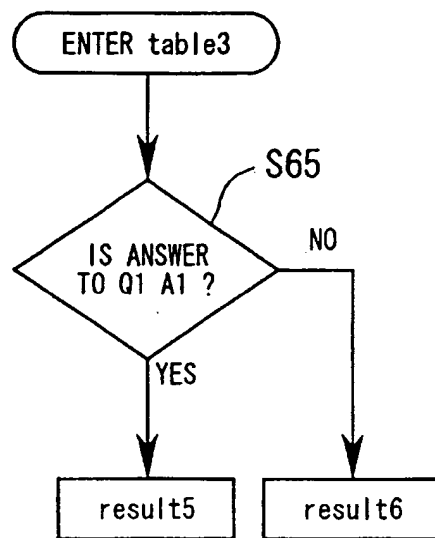


FIG. 19

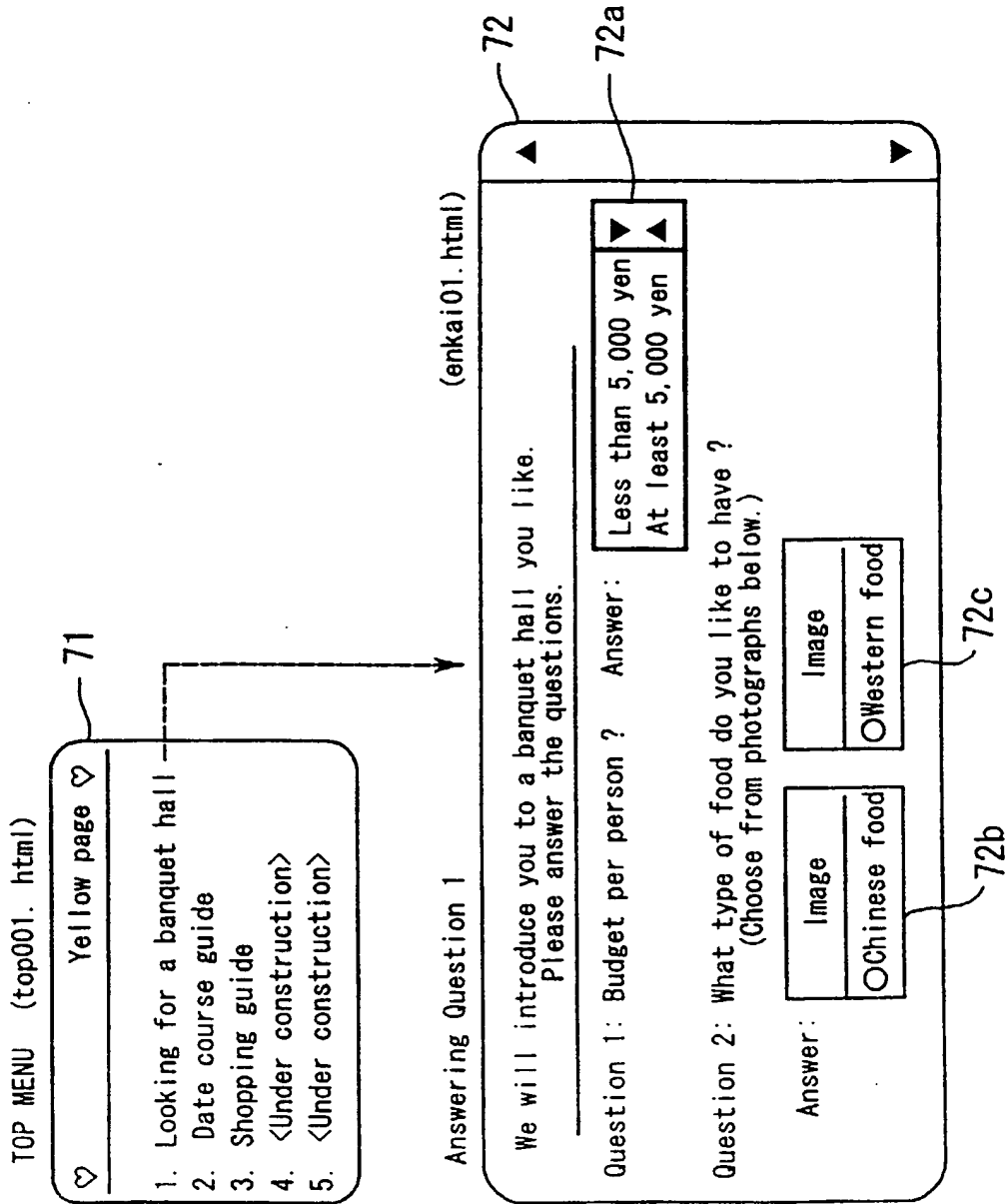


FIG. 20

Answering Question 2 (enkai01.html)

We will introduce you to a banquet hall you like.
Please answer the questions.

Question 1: Budget per person ? Answer:

Question 2: What type of food do you like to have ?
(Choose from photographs below.)

Answer:

Image
☐ Chinese food

Image
☐ Western food

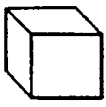
72b 72c

72a

73 (enkai14.gif)

Western food

Enlarged image



★♡○○▲

FIG. 21

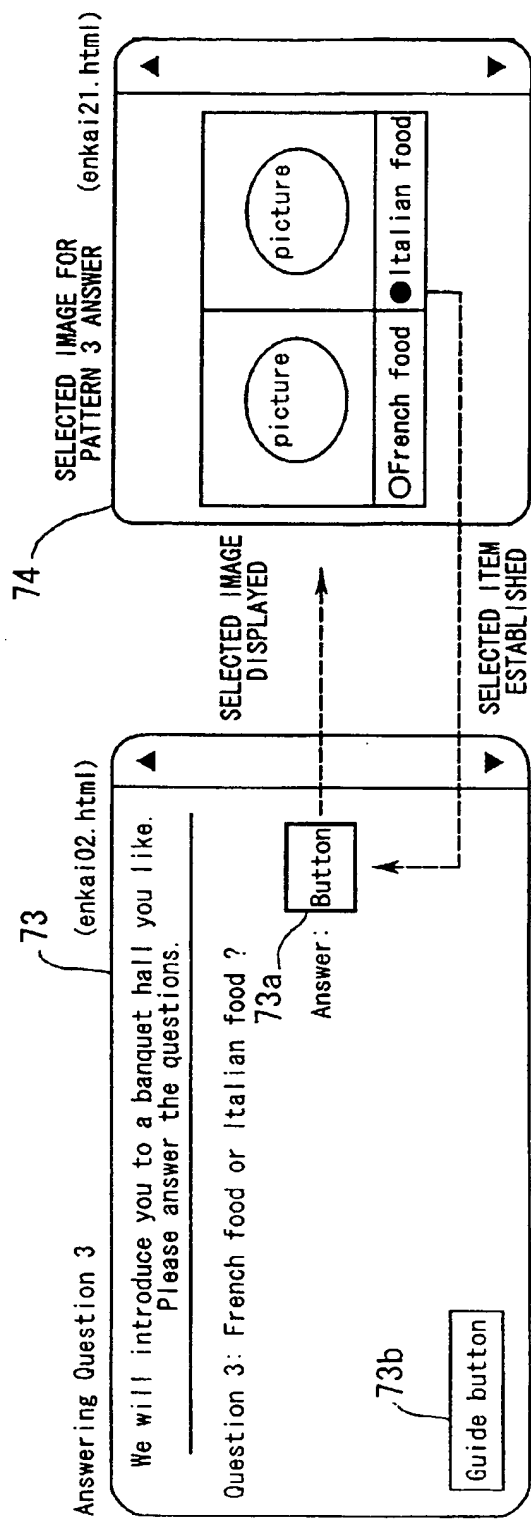


FIG. 22

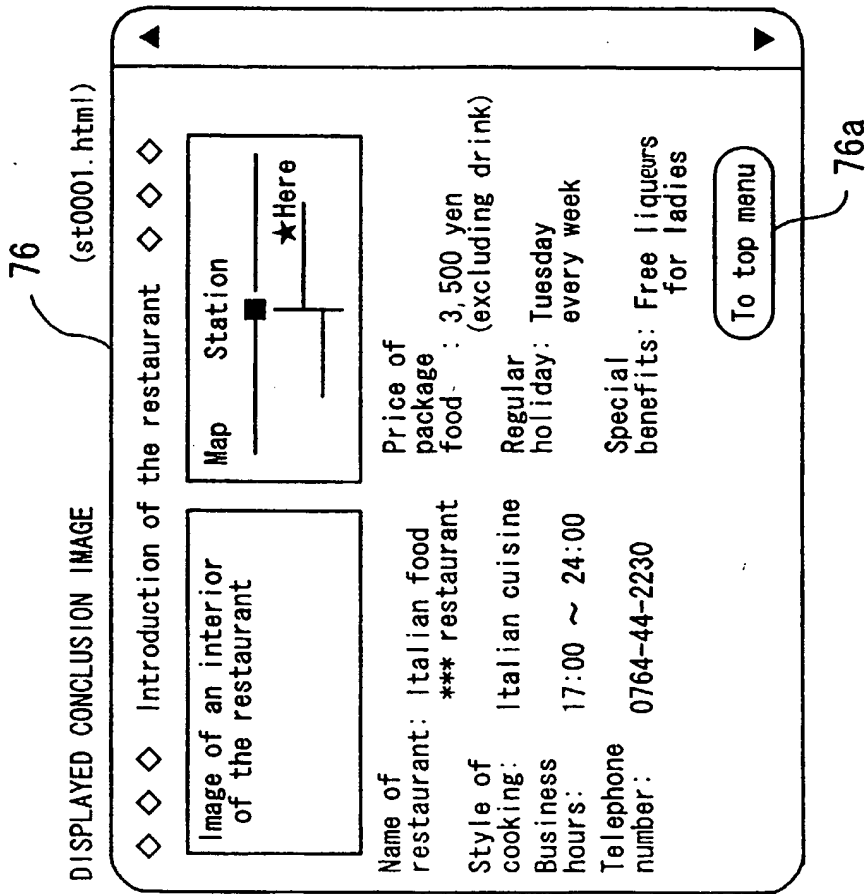


FIG. 23



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number

EP 97 30 6274

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.CI.6)
A	EP 0 314 279 A (IBM) * page 4, line 52 - page 7, line 30; figures 1-3E *	1,8	G06F17/30 G06F9/44
A	EP 0 578 830 A (ASAHI KASEI KOGYO KABUSHIKI KAISHA) * column 3, line 18 - line 41 *	1,6,8	
A	SCHULZRINNE H: "WORLD WIDE WEB: WHENCE, WHITHER, WHAT NEXT?" IEEE NETWORK: THE MAGAZINE OF COMPUTER COMMUNICATIONS, vol. 10, no. 2, 1 March 1996, pages 10-17, XP000580078 * page 14, left-hand column, line 48 - page 15, left-hand column, line 18 *	6,8	
A	BOERNER T: "DIE NEUE PROGRAMMIERSPRACHE JAVA REIF FUER DIE INSEL" CHIP ZEITSCHRIFT FUER MIKROCOMPUTER-TECHNIK, no. 3, 1 March 1996, pages 212-214, XP000557373 * page 214, left-hand column, line 30 - right-hand column, line 60 *	6	
			TECHNICAL FIELDS SEARCHED (Int.CI.6)
			G06F
The present search report has been drawn up for all claims			
Place of search BERLIN		Date of completion of the search 9 December 1997	Examiner Deane, E
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date O : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document</p>			

EPO FORM 1503 03 82 (P04C01)